ABSTRACT

ZHANG, QIANG. The Smart Agent-based Model in Urban Growth Problems. (Under the direction of Dr. Raju Vatsavai).

As per United Nations (UN) projections, it is expected that another 2.5 billion will be added to the urban populations by 2050. Urban growth models were widely employed to study urban expansion. In this study, we present a framework for the integration of an agent-based model (ABM) with the popular cellular automata (CA) based FUTure Urban-Regional Environment (FUTURES) model. This thesis addresses one of the key challenges in urban growth modeling, which is to capture the communication and behavior of agents in order to infer the agent's intent to develop a land parcel. With the interaction between agent-based model and urban growth model, we simulated the decision-making process of landowners when dealing with the urban development.

Many existing agent-based model frameworks were implemented using traditional shared and distributed memory programming models. On the other hand, recent Apache Spark is becoming a popular platform for distributed big data in-memory analytic. This thesis presents an implementation of agent-based sub-model in Apache Spark framework. With the in-memory computation, Spark implementation outperforms the traditional distributed memory implementation using MPI. This report provides (i) an overview of our framework capable of running urban growth simulations at a fine resolution of 30-meter grid cells, (ii) a scalable approach using Apache Spark to implement an agent-based model for simulating human decisions, and (iii) the comparative analysis of performance of Apache Spark and MPI based implementations.

The Smart Agent-based Model in Urban Growth Problems

by
Qiang Zhang

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Computer Science

Raleigh, North Carolina

2019

APPROVED BY:

Dr. Munindar P. Singh

Dr. Steffen Heber

Dr. Raju Vatsavai
Chair of Advisory Committee

DEDICATION

This is dedicated to my grandparents, Shuxiang Jiang and Jinyi Kang, who believe in me and inspire me, who love me and have supported me every step of the way.

"Cross the Bridge when you get to it, in the end things will mend." Without their comfort, I could never have walked this far.

BIOGRAPHY

Qiang Zhang received his bachelor's degree in telecom engineering in University of Science and Technology Beijing. After graduation, he joined a mobile security company and had been working as malware analyst for 3 years. He attended North Carolina State University in 2013, joining STAC Lab under the advice and Dr. Raju.

His thesis projects were focused on Agent-based model and high performance computation.

ACKNOWLEDGEMENTS

I would like to express my sincere, heartfelt thanks to my supervisor, Dr. Raju Vatsavai, for introducing me to the fantastic world of Geospatial Sciences, continually guiding me and giving me his invaluable advice, constant encouragement and precise modification on this road.

Besides my advisor, I would like to thank Center of Geospatial Analytics, for their insightful help and guidance of this work.

And also, my sincere appreciation to all of my friends, especially Ziqi Diao, M.J Mu and Binyuan Wu whom friendly encourage me in writing the thesis. When time gets tough, I know that you are with me and we have each other.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER

1

INTRODUCTION

In recent years, some UGMs have simulated the urban growth tendency. However, some of the work focuses on only one driver of urbanization [PF08] [Rom99] [KC03]. We believe that a realistic model of urban growth needs multiple driving factors and a means of modeling interactions between individual stakeholders in order to address the social factors involved in urbanization.

To achieve this goal of simulating urban growth to predict the area of urban development, agent-based models (ABM) could provide such advantages to present and model the urbanization growth with the multiple factors as economic status, individual consideration, and mass opinion. An agent-based model is a dynamic modeling method from the bottom up where all the individual computational elements are designated as an agent. The behavior of the system can be integrated from micro-level interactions of the agents. ABMs are widely used in various application domains, including but not limited to, biology, economy, psychology and crowd behavior studies. Simulations with ABM can obtain not only the result of the whole simulating system but also the snapshot of each agent in the system, which could help us understand the underlying relationships of the system further.

An agent-based model seeks to simulate the complex global behaviors via the most atomic units: individual, autonomous decision makers known as agents. The micro-level decisions involved in modeling agents manifest themselves globally in the aggregate behavior of the individual agents. We view this perspective as a natural way of modeling both the individual development decisions and the social factors involved. Besides, simulations involving ABMs allow aggregate behavior to be understood at any level of granularity since the underlying drivers are happening at the lowest level and not from the "top down." However, with the explosive growth of data involved in the simulation, more and more computational resources are required, especially in the geospatial application domain. In this paper, we build an ABM to simulate the decision-making procedure of the landowners in urban growth problems.

Our central goal in this work is to develop a discrete time-stepped intelligent agent model that couples urban growth with individual stakeholder decisions and takes demographic, economic and geographic information into account as driving factors. In addition, we also seek to demonstrate the application of machine learning algorithms in the modeling process. In our ABM design, we fully consider and conceptualize the possible principal elements that may affect the landowners' decisions in urban development. For a more accurate simulation, we use not merely the landowner's properties as computation input, but all the information of the cells and parcels which belong to the landowner is also taken into account as well. All the core characters are shaped into heterogeneous agents. In order to compare the computation performance, our module is built with both Apache Spark and a generic agent-based modeling platform, the FLAME, which supports message passing interface (MPI) and can be executed on multiple hardware and software platforms.

We make the following specific contributions:

- We implemented an ABM based on Apache Spark, which dramatically accelerates the simulation speed compared to the prevailing ABM framework the FLAME.

- We couple an ABM with a cellular automata (CA) model (FUTURES). The interaction between the two modules facilitates the ability of stakeholders to detect and analyze the change of the conditions. By doing this, we reduce the error propagation with better initial predictions.

- We implement an ABM to simulate the decision-making process in the urban growth process. Also, each agent in the system is given the ability to make an intelligent and more accurate decision by analyzing and understanding the historical urban growth patterns from the data over time.

- We specify a multi-level decision hierarchy instead of outputting yes/no decision for the stakeholders. Such design enhances the ability of stakeholders to make more flexible and intelligent decisions.

- We add a bargaining process in the urban development process between landowners and developers. Our results indicate that this strategy could better capture the stakeholders' behavior and conduct a more realistic simulation.

This thesis is mainly divided into two parts. In the first parts, we integrate the smart agent conception into existing urban growth simulation model, via this approach, we improve the model with more accurate prediction. And the second part covers an approach to make the urban growth simulation model execute faster by implementing Agent-based Model on Apache Spark platform.

CHAPTER

2

SAMRT AGENTS IN URBAN GROWTH SIMULATION

2.1 Related Work

CA models are used to represent the urban and regional growth along with many other technologies. [Ars13] proposed a hybrid model consisting logistic regression model, Markov chain (MC) and Cellular automaton (CA) to enhance the performance of traditional urban growth model. The research of [DS14] showed that the combination of (geographic information system) GIS and remote sensing (RS) could assist the CA-Markov model to predict urban growth trend of various land use classes. And [Fen16] also came up with a refined CA model which integrated partial least square regression (PLS-CA) and geographical information systems (GIS). The introduction of new algorithms and technologies could improve the accuracy of the urban growth prediction, but in some circumstances, people would like to understand the process of urbanization in a micro-level simulation. The urban growth and neighborhood influence were estimated through calibrated parameters, which reflect only the landscape features implicitly [CG98] [Cla97] [Cou97] [Bat07]. As a bottom-up model, CA cannot represent well macro-scale political, economic and cultural driving forces that influence urban growth [War00] [MS00] and [CH89] argued that the local economy has a significant impact on the urban growth, specifically mentioned the house price and land market.

Some existing CA models could be used to predict urban growth. While there are many similarities among the models, each still differs in its modeling framework and assumptions, as well as specific algorithms used to determine urban growth. So broadly these models can be categorized into deterministic and stochastic. GEOMOD [Hal95] [PJ01] is a relatively simple deterministic model, it needs the input as spatial data representing observed land cover, a site suitability surface, and an estimation of the quantity of cells to be developed. Here the site suitability is calculated through a regression function with the parameter of a pile of different layers of data. GEOMOD uses the optimization algorithm to select the most suitable location. SLEUTH [Cla97] is a much more complex deterministic model, and it is widely used through the United States, the input it requires are historical urban growth, road construction and slope along with an exclusion layer that acclaims what cells cannot be developed. LCM (Land Change Modeler) [Eas09] is a stochastic model, it has three modules: change analysis module to determine the quantity of cells to be developed based on historical urban growth data, transition potential model estimates a site suitability surface for new development, and change predictions that allocate new cells based on the site suitability surface. And FUTURES (FUTure Urban-Regional Environment Simulation) has three core components as Site Suitability, Per capita demand and Patch growth algorithm (PGA), And we use FUTURES as our CA model in our study, we will introduce more in later sections.

We did a comparison with these four models in Table 2.1.

Table 2.1 Model Comparison

	Flexible Input Data	User-Defined Quantity of Change	Probability Surface Required	Exclusive Layer	Stochastic Model	Stakeholder Influence
FUTURES	YES	YES	YES	YES	YES	NO
GEOMOD	YES	YES	YES	NO	NO	NO
LCM	YES	NO	YES	YES	YES	NO
SLEUTH	NO	NO	NO	YES	NO	NO

We can see that the common limitation is that the ignorance of stakeholder activity in the urbanization process.

Some researchers concern that human is the major determinant controlling factor. The work of [Ars13] and [LL07], they conceptualized three types of agents as the residence, developer, and governor. Each role of the agent has a corresponding social responsibility in the decision-making. Residence and developers and their interactions are the main force in the creation of a new urban area, and governors are the final decision maker. Similarly, [PF08]) and [Fil09] also conceptualized agents in LUCC problems, mostly from a perspective of economy. These researches addressed the interactions between supply and demand sides in term of economy, and the simulation of the bargaining process of the land transaction can be converted into the urban growth process when dealing with farmlands. In such cases, the representation of the economy in urban growth is crucial, and we can also decompose the problem with the analogue demand and supply measurements. Some CA models take account of the effect of the economy on the urban growth problem, but primarily focusing on the demand side of the process [Wat08] [SC02].

Applications of ABMs to study urban dynamics have increased steadily over the last twenty years, and researchers have studied the different models with different applied scenarios. Human decision-making procedure has also been taken into consideration with the agent-based model. The main advantage is an agent-based model can represent and observe the decision-making procedure from a bottom-up approach [Hua14]. Most of the study only comes with a study with the prediction precision [Mag11] [Mat07] [dâĂ02], however, with the increasing number of data and requirement of computation speed, the computational performance should be taken into consideration.

With the advent of Big Data, more and more application domains are trying to exploit and extract knowledge from this big data, ranging from global economy to society administration, and from scientific researches to national security [CZ14]. Some research shows the work of creating intelligent agents for making smart decisions in various application fields. [Yu09] used the fuzzy algorithm to analyze and evaluate the risk level of credit applicants. [Zhu08] applied cognitive map theory in route choice problem in travel demand modeling. And also some work has been done in ABM transportation and traffic simulation [Don08] [Par02] [CC10]. There are few studies the smart agents in the domain of urban growth.

2.2 Methods

We use a loosely-coupling framework to integrate the CA and ABM modules. Briefly, CA and ABM have its corresponding functions in the simulation, and they also communicate with each other to fulfill the goal.

ABM simulates stakeholders' behaviors and passes the final decisions of the stakeholders to FUTURES. Then FUTURES synthesizes the decisions along with landscape features to generate the urban growth simulation results and pass the growth update back to ABM. In our work, we define the interface between FUTURES and ABM in each time step and the logic flow of discrete time-stepped simulation.

In this section, we first introduce our study area and explain the data we collected. And then we present the methodology of the framework implementation.

2.2.1 Study Area

Cabarrus County is a region located in the north-east of Charlotte, North Carolina, USA, as shown in Figure 2.1. Since the late 19th century, Cabarrus has been an area of cotton cultivation and industrialization through cotton mills. Due to the economic development of the Charlotte metropolitan area, Cabarrus County has significant urban growth in recent decades. Despite the competitive advantage as being close to Charlotte, Cabarrus County has a limited size of large-scale manufacturing and agriculture is still leading a dominant role in its economy.

With the fact of mixed land use of agriculture and urban, Cabarrus County is an ideal region for the study of urban growth simulation.

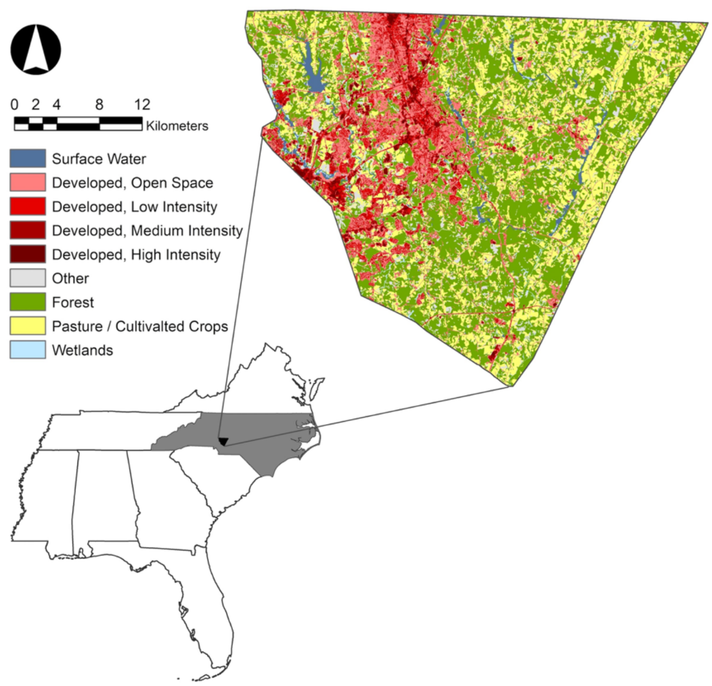

Figure 2.1 Cabarrus County, NC, USA

2.2.2 Data Description

To build an empirical model to simulate the urban growth in Cabarrus County, we collected the demographic data, GIS data, and land market transaction data for the ABM module.

We use the census age and salary data of the residence to represent the social and biologic status of the landowners. From the census data, we obtain the age and salary distribution of Cabarrus County and interpolate the data for each landowner from the distribution. Besides the census data, we also retrieved the land market transaction data of Cabarrus County from 2004 to 2016. The transaction data contains the land transaction history of the residential land use, farm land use, and commercial land use. We mainly focus on the residential and farm land use in the data.

Besides the demographic and economic parameters, we use GIS measurement to calculate the distance between each parcel to its closest green area and the highway. Distance to park could reflect the living comfort degree, the closer the green area is, the more leisure the residence could have, the fresher the air is. And the distance to the highway could present the travel convenience levels. Since there's no subway or many public transportation options in Cabarrus, the highway is the main transportation alternative. Therefore, we take but not limited to the distance to green area (Dist2GA) and distance to highway (Dist2HW) as the measurements of living comfort index.

We also collected the school rating scores of the primary, middle and high schools in Cabarrus County. According to the school district division, we assign each household a value of school rating based on the data we collected.

2.2.3 Framework Structure

We design an urban growth simulation framework that integrates an agent-based model to simulate the stakeholder involvement. In addition to the cellular automata urban growth simulation model, ABM plays an important role to simulate the behavior of stakeholders in the urbanization problem.

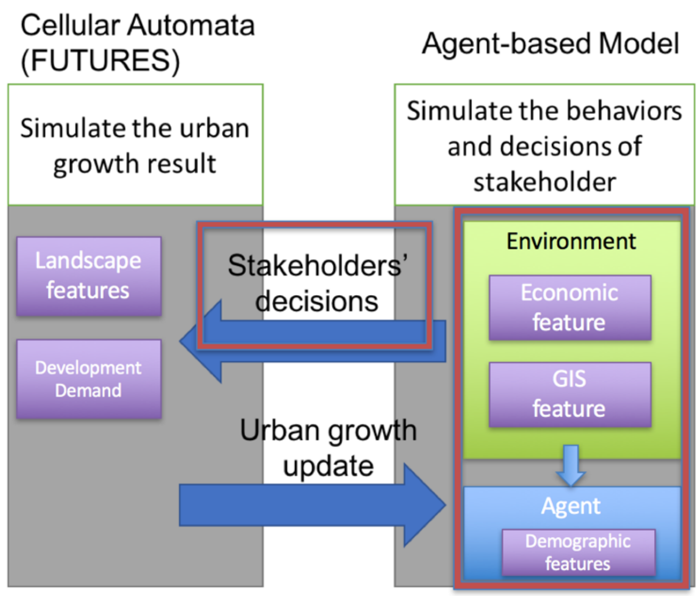

Figure 2.2 The communication between ABM and CA in each time step

As shown in the structure diagram in Figure 2.2, The framework is composed of CA and ABM, we use FUTURES as our CA model. As the initialization requirement of FUTURES, we initialize it with development demand and landscape features, such as slope, the distance to city center, and etc. Once acquiring all the initializing data, FUTURES generates the development potential DevPt of each landscape cell. Later FUTURES will use development potential to develop potential cells into the urban area.

On the other side, ABM has two separate parts, they are environment and agent sub-modules. Environment sub-module is responsible for collecting the data of the environment where all the agents live in. In our design, we collect two kinds of data in the environment sub-module, one is economic data such as land value, and the other one is GIS data such as the distance to green area. And agent sub-module is the main reactor of ABM, it has the ability to acquire the data from environment module and synthesizes its own demographic data, also it could respond to the development request with landowner's decision. All the data we collected in our simulation can be considered as the drivers to the urbanization process.

This design enhances the power of the framework to simulate a more empirical simulation. The selection of the drivers reflects the understanding of the problem modeling, and the implementation of each driver has a significant influence on the ultimate result of the simulation. In our study, we have age and salary drivers to assign age and salary for each landowner on the study area. And we also implement a land price estimator for each land parcel in the study area. We apply GIS tools to calculate the distance between each parcel to the nearest green area and highway.

Next, I will introduce more about the CA model in our framework, FUTURES.

2.3 FUTURES

FUTURES is a multilevel modeling framework for simulating changing regional urban-rural divide based on observed land change patterns [Mee13]. It is a useful tool for researchers of urban systems to study the effect of changing landscape patterns with urbanization.

FUTURES considers a variety of environmental, infrastructural, socioeconomic and demographic factors as inputs to build a mathematical model representative of observed

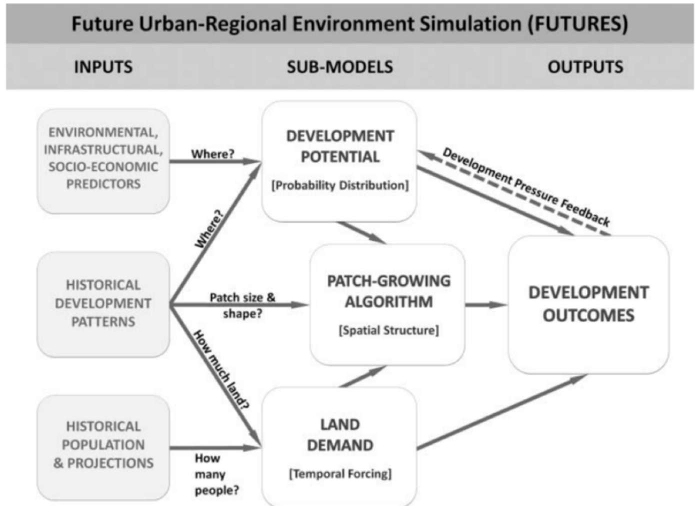

Figure 2.3 The FUTURES land change modeling framework [Mee13]

land growth patterns. These inputs are non-stationary drivers of land change influencing the spatial structure of new growth regions and known to vary with time. Also, FUTURES also considers simulated land conversion events in a feedback loop to predict new land change patterns. To incorporate the different drivers of land change, the spatial structure of land change events and their influence on further conversions, the FUTURES framework couples three interacting components: a DEMAND sub-model that projects the estimated demand for land in a region, a POTENTIAL sub-model that quantifies the site suitability of land in a region and, a patch growing algorithm (PGA) that produces projections of regional landscape patterns based on outputs from the DEMAND and POTENTIAL sub-models.

Figure 2.3 provides an overview of the FUTURES simulation framework and its interacting components, namely, (i) DEMAND sub-model, (ii) POTENTIAL sub-model and, (iii) Patch Growing algorithm (PGA)[Sha16].

The POTENTIAL sub-model implements a site suitability modeling technique that formalizes the relationship between urban development and environmental, infrastructural, and socioeconomic changes over time in a region. The model considers a number of predictor variables as input to a multilevel logistic regression model. Each predictor variable accounts for a spatial or temporal aspect of land cover change and is used to define a suitability score for a site in a region. Finally, the output from the POTENTIAL model is normalized to produce a map of development probability values for all sites in a region.

The probability that an undeveloped cell becomes developed is defined as:

$$p_i = \frac{e^{s_i}}{1 + e^{s_i}}$$

where S_i is the composite development potential for a cell i. The development potential S_i is defined as a function of environmental, infrastructural, and socioeconomic predictor variables of site suitability as following:

$$S_i = a_{ji} + \sum_{h=1}^{n} \beta_{jih} * x_{ij} + \beta_{jih} * p_i'$$

where, for the $i^t h$ undeveloped cell and varying across j groups (i.e., the level), a_{ji} is the intercept, β_{ji} is the regression coefficient, h is a predictor variable representing conditions at the start of a chosen simulation year, n is the number of predictor variables, x_{ih} is the value of h at i, and p' is the dynamic development pressure variable due to neighboring developed sites.

$$p_i' = \sum_{k=1}^{n_i} \frac{State_k}{d_{ik}^{\gamma}}$$

where $State_k$ is the current state (0/1) of the $k_t h$ neighbor of cell i, d is its distance from the $k_t h$ neighbor in its list of n neighbors and γ is a coefficient that controls the influence of distance between cell i and its neighboring cells.

The DEMAND sub-model establishes the relationship between historical land consumption and population growth under different development scenarios. This relationship is established using an ordinary least squares regression technique. The regression model considers two parameters in estimating future land use for urbanization: (i) population growth and (ii) land consumption over time. The generated per capita demand projections from the model drive the PGA in the FUTURES UGM.

The Patch Growing Algorithm implements the mechanism to simulate historically observed urbanization patterns based on the above two sub-models. Patch growth is defined as a 3-step process: (i) Monte Carlo based seed selection using the site development probability, (ii) patch size selection from a library of patch sizes, a weighted distribution of historically observed patch sizes, and (iii) patch growth through a neighbor discovery process.

A neighborhood configuration specified at the start of the simulation defines the neighbor discovery process. Further, a site suitability metric for newly discovered cells determines the most suitable cells that go towards patch growth.

$$s_i = s_i' * d^{-\alpha}$$

where s' is the underlying development potential of cell i, d is the distance of cell i to seed and α is a patch compactness factor. The PGA continues the neighbor discovery process by using the newly added neighbors as potential seed cells. Thus, the patch growing algorithm continues till the value of patch size is met or else terminates when no more suitable sites for patch growth can be found.

2.4 Model Drivers

The initialization of the framework requires empirical input, this section introduces the data preparation.

2.4.1 Demographic Data

For this study, we collected the Cabarrus County demographic data like age and income of the household. According to the census data, we built distribution samplers for age and salary. We use a normal distribution for each age group of the landowners. The census data provides the number of people in each age group within each street block. We assume the age of people falls into the normal distribution. Based on the age range and percentage of each age group, we assign the landowner an age, and the age increased by one with the simulation continues over years. We also simulate the death of a landowner. We assume that after the age of 79 (US average Life expectancy), the landowner has a chance to survive, otherwise, we replace a new landowner based on the age and salary distribution.

And we use a skewed distribution, chi-square distribution, to describe the salary distribution over the landowners. As the feature of chi-square distribution, we use the median value of the salary of each age group within each street block to obtain the degree freedom, as the equation below.

$$median = k(1-\frac{2}{9*k})^3$$

Here, k is the degree freedom of the distribution, and according to the survey conducted by Society for Human Resource Management, the average annual salary raise is about 3.1% per year. So in our implementation, we assume that the salary for each landowner could increase by a fixed rate every year.

The landowner could be replaced as a result of two reasons, one is the death of a landowner, and the other is the land trade transaction. Either of the above occurs, a new landowner will replace the old one. And a new landowner will occur when a new parcel is developed.

2.4.2 Land Market

The land market is a crucial factor in our study, the prediction of the land price could affect the demand and decision-making of the stakeholders. We built a land price estimator based on the historical land price to estimate the land value over years for every parcel on the map.

While some researches have focused on factors that may affect the land value of a region, such as the human density and the distance to central business district; however, the data we obtained is too scarce to implemented any existing complex land value prediction model [McM96]. Therefore, we selected a straightforward method to estimate the land value based on the available data, and test the approach with the statistics data, which compare the median value of the predicted land values with the actual land values distribution.

Obviously, not all the parcels were sold in the past years, limited to the scope of the transaction data of the land trade, we need to interpolate the data for every parcel. We use the K-nearest-neighbor algorithm to interpolate the land price for each year. For example, if we want to know the land value of a parcel for a certain year, we locate the nearest neighbors (geographical distance) which had been sold in the same year and calculate the average value of its neighbors for the demand parcel. With this method, we calculated the land value for all the parcels on the map.

Table 2.2 Comparison of median value of land (per sqrt feet) value distribution before and after the interpolation

zipcode	County	2013BM	2013AM	2014BM	2014AM	2015BM	2015AM	2016BM	2016AM
28027	Concord	86.41	82.32	90.67	89.77	95	92.23	100.67	96.7
28025	Concord	81.83	80.78	85.25	83.78	89.17	89.67	94.58	92.3
28081	Kannapolis	73.42	70.45	76.17	76.23	79.83	80.89	85	82.67
28083	Kannapolis	70.5	71.67	72.5	70.73	75.25	72.23	80.42	76.8
28075	Harrisburg	93.5	87.23	97.5	98.67	101.75	103.5	106.92	110.21
28107	Midland	99.33	95.45	102.3	98.98	106.17	104.56	112.08	105.4
28124	Mount Pleasant	93.41	93.56	97.75	94.4	100.17	100.4	104.33	102.7

BM is the median value of the land value distribution before interpolation, and AM is the median value of the land value distribution after interpolation. From Table 2.2, we can see the tendency of the land value for each zip-code in Cabarrus County. With the development of economy and urbanization, the land value is increasing by years. And we can also observe that the land value declines with the distance to Charlotte (in the South-western of Cabarrus County).

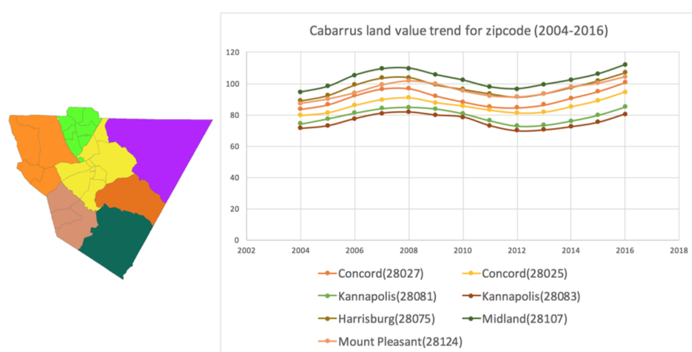

Figure 2.4 Cabarrus land value trend for zip code (2004-2006)

From Figure 2.4, we can clearly observe the tendency of the land value on Cabarrus County, and the tendency of different zip-code area follow almost the same pattern. In order to estimate the land value in the future, for each parcel, we use linear regression to predict the land value based on the historical values. By interpolation and prediction of the land value, we considered both spatial and temporal change of the land value, and the historical land value could be well interpolated across the map.

2.4.3 GIS Data

GIS data is also widely used in the urbanization problem [CG98] [Bat07] [LY00] [Sud04], and the application of GIS in urban growth prediction propel the implementation of the models. We use multi-layer data, GIS is applied as a data integrator. Although the data is already existing, we created new maps to fit the grid resolutions and projections.

And the other purpose of using GIS is visualization. This is the weak component for urban growth, GIS can just enhance the model. The role of GIS in for visualization is taking the intermediate result and generate the visible map for the comparison of different maps.

The model we built requires the DIST2GA and DIST2HW as inputs to ABM, we projected the shapefile of Cabarrus County transportation and Cabarrus County green land onto the Cabarrus County land parcel, generating the raster file of the distance from each parcel to its nearest highway and green area. With the same method, we also mapped each parcel to corresponding schools (primary, middle and high schools), and calculate the average school rating score for each parcel.

Up to this point, we prepared the initial data to train and test ABM, as listed as (i) landowner age, (ii) landowner salary, (ii) land parcel distance to green area, (iii) land parcel distance to highway, (iv) land parcel value, (v) land parcel school rating. With the above data, we designed our discrete-time agent-based model to simulate the decision-making of the stakeholders.

2.5 Agent-based Model

To interact with urban growth model, the human involvement simulation should be capable of responding with the landowners' decisions while retrieving sufficient information for both the internal(landowner) and the external(environment) properties. With more information considered, a simulating module could always generate better predictions. In our design of the agent-based model, we fully considered the heterogeneity of the model, each agent on the map is an autonomous individual [Zha16]. The system level behaviors emerge from the micro-level interactions of the agents. Therefore, it could depict the whole simulation procedure with the details of different levels, specifically reasoning how and when a decision is made by the landowners. Also, ABM aggregates all the information of its constituent elements and measure the outcomes in the system level over time.

2.5.1 Agents Conceptualization

We conceptualize the model obeying ODD protocol [Gri10]. The ODD protocol is a standard format proposed to describe the individual and agent-based model.

The ABM is designed with two-layers, one is landowner layer, and the other is environment layer. As illustrated in Figure 2.5, available parcels (Not all the parcels are legitimate for developing, such as roads) and all the landowners on the layers are conceptualized as agents, which maintains the information passed by the model drivers. With the model driver updating the values, the agents update its memory.

When the simulation starts, the landowner retrieves the land parcel information where he/she owns. Landowner as the decision maker, he/she gathers all the information needed to make the decision. And the reasoning rules are obtained via the training of the historical data. In our experiment, we tried various kinds of machine learning algorithms to test their performance in this framework.

The key outcome of the model is the decision map, indicating if the landowner is willing to sell some certain landscape cell to develop. And the FUTURES is seeking the objective that the landowner is willing to sell this parcel based on the decision map generated by ABM, and in return, the newly developed parcels trigger ABM generate new landowner agents.

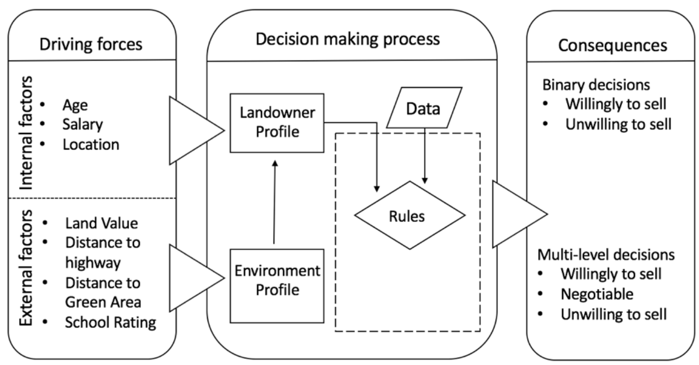

Figure 2.5 Agent-based model conceptualization

The outcome of ABM synthesizes not only the landowners' condition but also their neighbors' impact on them. Besides the agents internal updating, we also modeled the influence of external agents of the experiments. In the experiment, we also tried different mechanisms for different scenarios.

2.5.2 Agents Decision Rules

Traditionally, the decision of the landowner is simply yes or no, such implementation sometimes leads to the insufficiency of cells to be developed. This is because landowners make a decision based on the decision rules, and there's also a development demand for each year, this problem happens when the number of the willing landowner is less than the demand. To solve this problem, we proposed a new decision hierarchy to replace the binary yes/no. In our design, the decision is 3 categories as willingly, negotiable and unwillingly. When there are not enough âĂIJwillinglyâĂİ landowners, we will convert some negotiable landowners into âĂIJwillinglyâĂİ ones by the process of bargaining. And the simulation could continue with enough cells to develop.

We conclude the decision rules with some machine learning algorithms. For binary decisions, we use the historical land transaction data to train our model. From the transaction record, we can tell which cell is sold in which year, combining with other features which we have initialized for each year, we can try to find the decision-making pattern. With logistic regression algorithm, we can obtain the function with the parameter of age, salary, distance to green area, distance to highway, land value and development potential.

When it comes to multi-level decisions, we cannot get such label from the historical transaction record directly. We use the time interval between two transactions of the same cell to determine the decision labels. When a piece of land is sold within a certain threshold, we can infer that when a person with certain characteristics owns a piece of land, he is more likely to sell it. The threshold we use is the average moving time of US residence, it is 5 years. So we define that when the interval is less than 4 years, the stakeholder is more willing to move, and longer than 6 years is unwilling to move. Any other time between 4 and 6 years, which contains 5 is negotiable to move. We test different classification algorithms to train our model with multi-level decisions, the algorithms are CART [Bre17] [FB97], Random Forests [LW02], SVM [CV95] [Yan08], and Bayesian networks [Jen96]. we select the best one to fit in our model to simulate the decision-making process of the landowners.

2.5.3 Agents Neighbor Influence

We consider the neighborhood influence among agents. And For a different decision setting, we have different influence calculation methods. For binary decision, since we are using logistic regression, we find a way to aggregate itself value and its neighbors. For multi-level decisions, we define a cross-reference table for neighborhood influence as shown in Table 2.3. In the neighborhood, we apply majority voting, selecting the most popular decision and consider both the local and neighbor decision, according to the table, we have the final decision.

Table 2.3 Neighborhood Influence for multi-label approach

	Willingly	**Negotiable**	**Unwillingly**
Willingly	Willingly	Willingly	Negotiable
Negotiable	Willingly	Negotiable	Unwillingly
Unwillingly	Negotiable	Unwillingly	Unwillingly

2.5.4 Agent's Bargain Process

We implement multi-level decision because of insufficient cells to develop in the urbanization process. And in an empirical study, developers would like to negotiate with the landowners or providing them a higher bid [PF08]. When adopting the multi-level decision, we need to consider converting negotiable landowners into willingly ones, and in the meantime, the land value should increase by a certain percentage. The increasing rate is determined by an experiment which will be introduced in the experiment section.

2.6 Experiments

We split the experiment into four parts, the first part is to train the model to generate the decision-making rules for the landowners, to test under what circumstances that they would sell their land. The rest of the experiments are all conducted in the form as the comparison of 3 platforms as the plain FUTURES, the FUTURES with binary decision ABM and the FUTURES with the multi-level decision. And the second experimental part is to simulate without neighborhood influence, and the third experimental part is to simulate with neighborhood influence. The fourth experimental part is to simulate with neighborhood influence and the bargaining process. Additionally, before the fourth experiment, we need to determine the land value increase rate.

All the experiments were initialized with different scenarios of different landowner age and salary assignment. For each scenario, we repeated the experiment 10 times, and we summarize all the result to get an average value for validation. The simulation year is from 2014 to 2016.

2.6.1 Validate Metrics

We used two validation metrics in the result analysis of our experiment, they are allocation agreement and willingness agreement. Allocation agreement indicates the precision of the urban growth simulation prediction, representing the ratio of correctly predicted developed cell status.

As illustrated in Figure 2.6, the allocation agreement is the ratio of the overlapping developed parts of the truth and simulation to the total number of cells on the map.

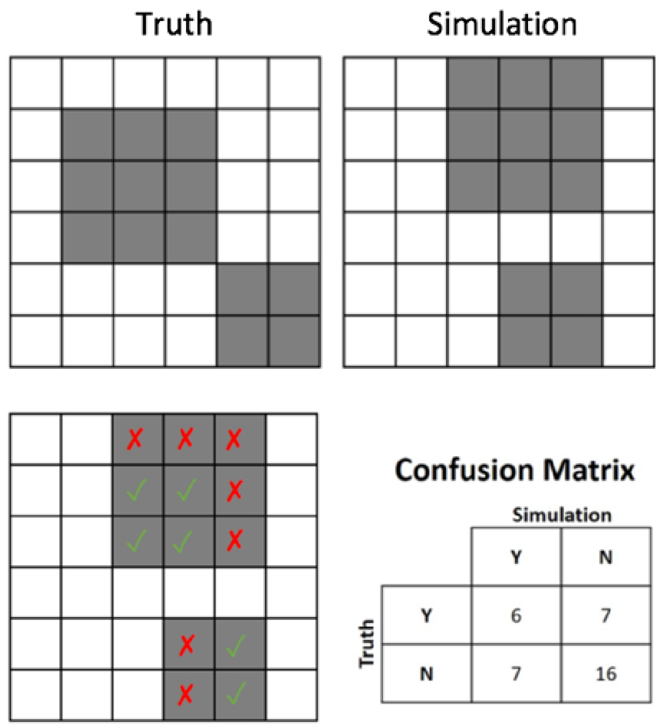

Figure 2.6 Allocation agreement illustration

$$AllocationAgreement = \frac{Num(TP)}{Num(TP)+Num(FP)+Num(TN)+Num(FN)}$$

And the other metric is willingness agreement. Willingness agreement is the measurement for the performance of the agent-based model, representing the ratio of correctly predicted decisions. Since we don't know the actual decisions of the landowners, we could only imply that via the actual development map. We compare the predicted decision-making result with the actual development map. As shown in Figure 2.7, we can only retrieve the false negative in the confusion matrix, and we are not sure about others. So, we defined the willingness agreement as:

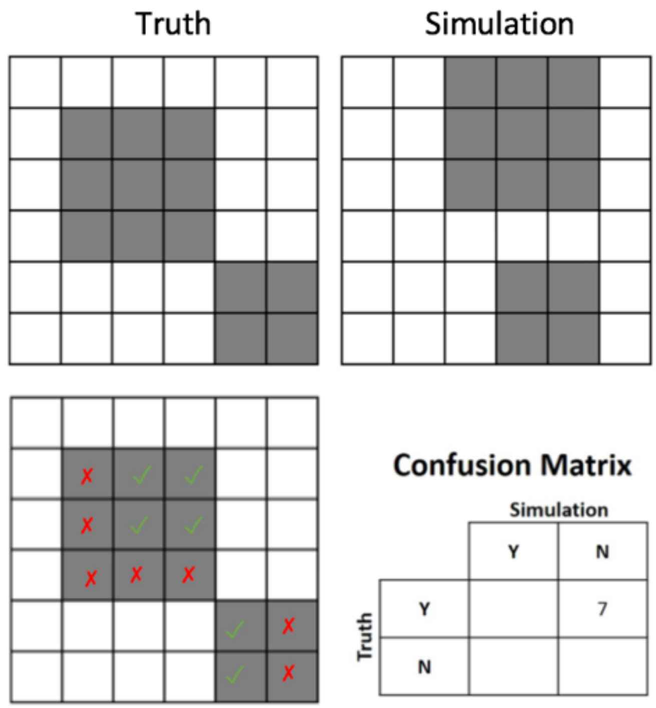

Figure 2.7 Willingness agreement illustration

$$WillingnessAgreement = \frac{Num(FN)}{Num(TP)+Num(FP)+Num(TN)+Num(FN)}$$

2.6.2 Train the Model

We obtained the historical transaction data for each parcel, we use this dataset to train and test the performance of different machine learning algorithms. Along with this dataset, we assign an age and salary to the landowners based on the model drivers, the DIST2GA and DIST2HW to each, and the sold price (per sqrt feet) from the record, and also generate the development potential for the parcel in corresponding year from FUTURES. Thus, for each item in the transaction record, we have the landowner age, landowner salary, DIST2GA, DIST2HW, land value, development potential and the sold status, which is the decision of the landowners. We regard the other parcels which are not listed in the record as unsold.

For this raw and extended dataset, we have 59136 sold records and 1350740 unsold records, and we use the data record from 2004 to 2013 as training data, and the rest as testing data.

We use logistic regression [KZ01] [BH03] to generate binary output as the decision of the landowners. Table 2.4 is the result of the test:

Table 2.4 Logistic Regression result

	Precision	**Recall**	**F1-score**
Sold	0.66	0.74	0.70
Unsold	0.71	0.62	0.66
Avg/Total	0.69	0.68	0.68

We calculated the intervals of transactions of each parcel, retrieving the multi-label decision of the landowners.

Table 2.5, Table 2.6, Table 2.7, Table 2.8 are the results of different algorithms:

Table 2.5 CART result

	Precision	**Recall**	**F1-score**
Willingly	0.74	0.73	0.74
Negotiable	0.70	0.70	0.70
Unwillingly	0.58	0.62	0.60
Avg/Total	0.71	0.70	0.71

Table 2.6 SVM result

	Precision	Recall	F1-score
Willingly	0.54	1.00	0.70
Negotiable	0.99	0.23	0.37
Unwillingly	1.00	0.27	0.42
Avg/Total	0.60	0.60	0.53

Table 2.7 Bayesian Network result

	Precision	Recall	F1-score
Willingly	0.51	0.59	0.54
Negotiable	0.45	0.43	0.44
Unwillingly	0.28	0.14	0.19
Avg/Total	0.46	0.47	0.46

2.6.3 Simulation

We chose random forest as the multi-label algorithm to compare with binary output result since it has the moderate but steady performance among the competitors.

we took 2013 as a starting year and test the development status of 2014, 2015 and 2016. We did three simulations with different conditions, we tested the framework with the plain frameworks, FUTURES model alone, FUTURES model with binary decision ABM and FUTURE model with multi-level decision ABM. Firstly, we assume that the landowners are not affected by its neighbors (adjacent parcels) and no bargaining process involves. And in the second experiment, we added the neighborhood influence in the framework. And

Table 2.8 Random Forest result

	Precision	**Recall**	**F1-score**
Willingly	0.74	0.76	0.75
Negotiable	0.70	0.71	0.70
Unwillingly	0.66	0.59	0.62
Avg/Total	0.78	0.72	0.72

last, we added the bargaining process when dealing with multi-level decision ABM model. Before the simulation, we need to determine the optimal land value increase rate of the bargaining process. As shown in Figure 2.8, we increase the rate from 5% to 15% by the interval of 0.5%. And the result shows that with the rate of 8%, it gives the best simulation result.

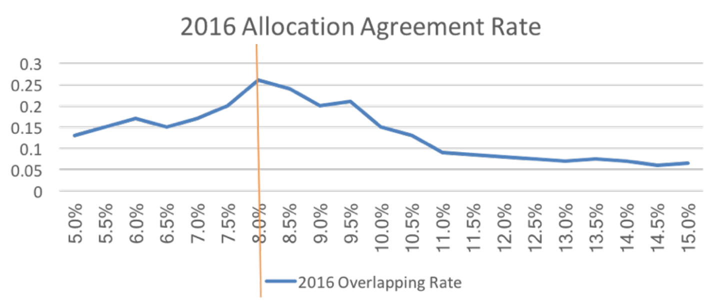

Figure 2.8 Prediction accuracy with land value increase rate change

The experiment results are shown in Table 2.9, Table 2.10, Table 2.11:

Table 2.9 Simulation Result without Neighborhood Influence

Year	Framework	Demand	Successfully Developed	Allocation Agreement	Willingness Agreement
2014	Plain Futures	9773	9773	17%	-
2015		9773	9773	15%	-
2016		9773	9773	14%	-
2014	Binary Decision	9773	9773	18%	35%
2015		9773	8750	17%	46%
2016		9773	-	-	-
2014	Multilevel Decision	9773	9773	24%	44%
2015		9773	6164	20%	60%
2016		9773	-	-	-

Table 2.10 Simulation Result with Neighborhood Influence

Year	Framework	Demand	Successfully Developed	Allocation Agreement	Willingness Agreement
2014	Plain Futures	9773	9773	16%	-
2015		9773	9773	16%	-
2016		9773	9773	14%	-
2014	Binary Decision	9773	9773	20%	40%
2015		9773	7896	19%	45%
2016		9773	-	-	-
2014	Multilevel Decision	9773	9773	29%	60%
2015		9773	6164	22%	62%
2016		9773	-	-	-

Table 2.11 Simulation Result with Neighborhood Influence and Bargain

Year	Framework	Demand	Successfully Developed	Allocation Agreement	Willingness Agreement
2014	Plain Futures	9773	9773	17%	-
2015		9773	9773	16%	-
2016		9773	9773	15%	-
2014	Binary Decision	9773	9773	22%	48%
2015		9773	7654	19%	45%
2016		9773	-	-	-
2014	Multilevel Decision	9773	9773	32%	60%
2015		9773	9773	24%	62%
2016		9773	9773	26%	68%

2.7 Results and Analysis

From the results of the experiment, on average, the binary and multi-level framework outperform the plain model of FUTURES alone for both the overlapping rate and disagreement rate. And in the comparison with binary and multi-level decision framework, multi-level decision framework performs better. In the first and second experiment, multi-level approach failed in development enough parcels, so the land growth model quit in developing. The reason for that is there are not sufficient landowners agree to sell their land, and the developers do not negotiate with these landowners. The overlapping rate of the multi-level approach is lower than the third experiment, one of the reasons is that the fewer number of the cells were developed. However, the disagreement rate is lower than the other two approaches in the first experiment.

When adding neighborhood influence, we noticed that the performance of all has been improved, and the multi-level approach developed more cells than in the first experiment.

That indicates that when the interaction between landowners occurs, the landowner learns more of the environment through his/her neighbors, which helps the landowner make the correct and wise decision.

In Table 2.10, Multi-level approach met the demand of development with moderate performance. When the bargaining process involved, we can see that the allocation agreement rate is raised, which verifies our assumption that neighborhood influence and bargain process could improve the landowner agents' ability to make more intelligent decisions. And the land value increase rate indicates the commercial pattern in land trade. When the value of the land is low, fewer landowners would like to sell their land; and when the land value is too high, the allocation agreement rate drops because of fewer developers would like to buy and develop a new site.

Besides the spatial error estimation, temporal error estimation is also important. Error propagation exists in the simulation. We did another experiment to test the impact of error propagation that small initial error in geo-simulation will lead to larger errors in the end. We manually inserted incorrect data into our initial landscape file. We randomly selected 10%, 15%, and 20% cells on the map, transposing their development status. With such preparation, we intentionally aggrandized the initial error in order to observe the impact of error propagation.

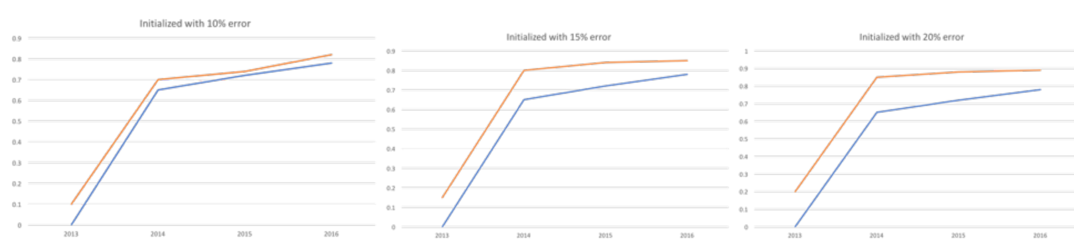

Figure 2.9 Error propagation observation

CHAPTER

3

BUILD ABM ON APACHE SPARK

3.1 Problem Statement

Urbanization is a process that accumulates for a long time. Therefore, a significant change of the land use could take place after years. In the prediction of the urban growth, researchers usually measure the land use status of the past years and simulate the following land use status of the next decades.

Many factors would affect the urban growth and development, including the properties of the landscape and landowners. For a better illustration of the algorithm and model, we define two basic spatial landscape units.

Landscape cell: We grid segment the landscape, and each grid of the landscape is called a cell.

In many spatial problems, satellite images are used for further analysis. We can regard each pixel of the image as a cell. And in some actual landscape maps, we use such representation to convert continuous data into discrete data.

Landscape parcel: A group of adjacent cells which belong to the same landowner is call a parcel.

A landowner may have more than one parcel on the landscape, and all of the parcels from the same landowner may sparsely locate.

We also define one spatial relationship of the landscape parcels.

Parcel neighborhood: A group of parcels which are adjacent to a parcel, is called this parcel's neighborhoods.

For instance, we assume there is one piece of the landscape taken into consideration, as shown in Figure 3.1. Each color area indicates a landscape parcel, and each parcel is composed of multiple cells, as represented by dotted squares. For better illustration, we mark each parcel a name alphabetically. As defined above, only the adjacent parcels are considered as the neighborhood of a particular parcel. In this example, parcel A's neighbors are C, D, E, F, G, H.

Hence, we have an overview of the landscape units and the relationships between them, and we would specify the problem we are facing. Human involvement is inevitably in urbanization, and people have the ownership of the land and the willingness to trade. Also, one person's decision could be affected by others, like peers or neighborhood. We need a module simulating and monitoring human involvement and conduct further qualitative and quantitative analysis.

This human involvement simulating module should take the landscape and landowner

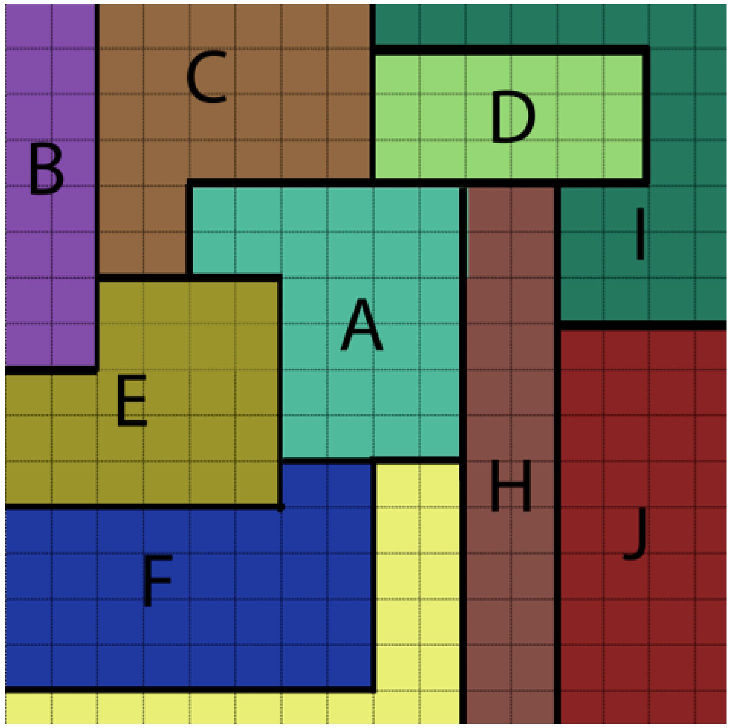

Figure 3.1 Example of landscape cell, landscape parcel and parcel neighborhood

information as input, including the landscape cells' position, landscape parcel composition, landscape parcel ownership, the profiles of landowners and other landscape information whichever used in the decision-making algorithms. With the information of landscape and landowners, this module could calculate the probability of how likely the landowner is willing to sell a certain parcel and make the ultimate decision of the landowner. Since this module is working with the urban growth simulation model, it could respond to the urban growth model when it's requested to provide the final decision if any parcel is going to be sold or not.

Since the urban growth simulation is usually done for decades in the future, researchers usually simulate not for a single year, but a series of continuous years. In our study, we simulate the urbanization for a 20-year period. Figure 3.2 shows the procedure of a multiple-year simulation as working with both an urban growth simulator (UGS)and a human involvement simulator (HIS). UGS waits for the feedback from HIS, the response time of HIS is the gap of the overall performance; we aim to reduce the response time of HIS to improve the performance of the simulation.

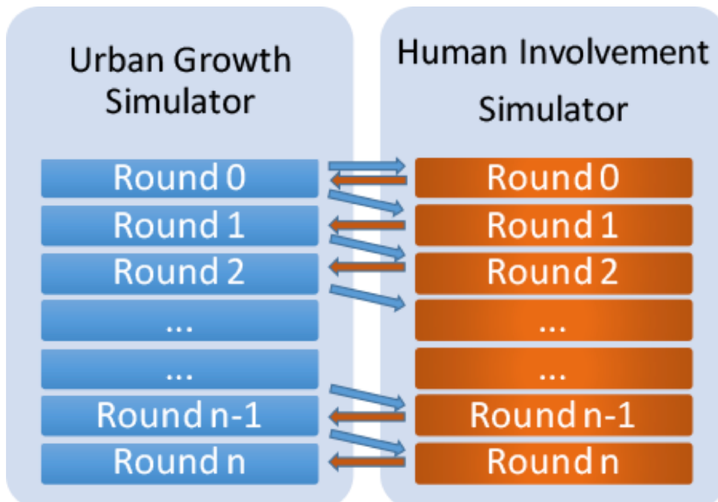

Figure 3.2 Communication between Urban Growth Simulator and Human Involvement Simulator for Rounds

Building a stable and scalable human involvement simulating module can assist the urban growth model to make a more accurate prediction, and help people analyze how human affect the urbanization procedure.

3.2 Methodology

To interact with urban growth model, the human involvement simulating module should be capable of responding with the landowners' decisions while retrieving sufficient information from the landowner and landscape. With more information considered, a simulating

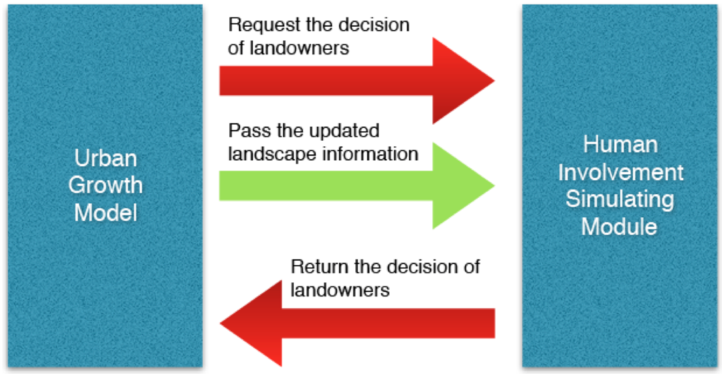

Figure 3.3 Human involvement module communicates with urban growth model

module could always generate better predictions. In our study, we pass the landowners' profile and landscape geographic information as an initial input to the human involvement simulating module; some decision-making algorithms could convert the data into landowner's decisions.

3.2.1 Formulas

We consider landowner's the probability of the landowners' willingness to trade their land is based on the parcel, which means we assume that all the land trades are in the unit of a parcel. However, all the landscape information we gathered is based on landscape cells, we need algorithms to transfer the landscape information from cell level to parcel level. Before introducing the algorithms, we need to define some critical parameters we used in the algorithms.

Development potential: Development potential is a value indicating how likely a cell would be developed into an urban area without human involvement.

While urban growth model is working in a standalone mode, development potential is the decisive factor to change the status of a landscape cell or parcel. The value is calculated with the innate and acquired conditions of the landscape cells, as well as the adjacent cells' information. Development potential synthetically considers the interior and the exterior of the landscape conditions.

Willingness to sell: Willingness to sell (WTS) is a value indicating how willing a landowner would be to sell the landscape cell or parcel.

When human involvements occur, the intention of the landowners becomes crucial. In the trade of one piece of land, landowners always synthesize multiple factors and make the final decision if the land could be sold or not. And during the decision-making procedure, a landowner could also be influenced by other landowners. Due to the different personalities of the landowner, their behavior can be varied in pondering the decision. Overall possible influencing factors, we use WTS to represent the possibility a landowner could sell a certain cell or parcel.

As shown in Figure 3.4, a cell's WTS should be calculated based on the landowner information and its development potential, the algorithm we use to calculate cell's WTS is illustrated as below.

Let WTS_{cell} to be the willingness to sell value of a landscape cell, and potential to be the development potential value of this landscape cell. If a landowner owns this landscape cell, we assume that

$$WTS_{cell} = intersection + \alpha * age + \beta * income + \gamma * potential$$

where $intersection$ is a value that varies among different types of landowners, age is the age of the landowner, and $income$ is the annual income of a landowner. α, β, γ are the coefficients of the parameters. In our study, we use α as 0.0032, β as 0.00000486, and γ as 0.869.

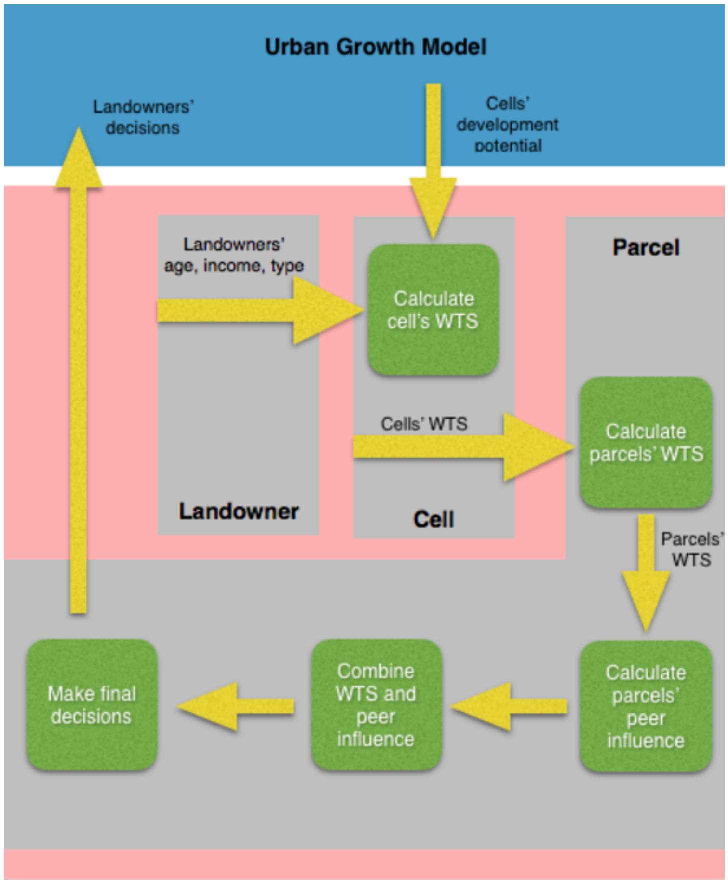

Figure 3.4 Working flow of urban growth model and human involvement simulating module for one-time step

Since our study is based on landscape parcels, we should obtain the intention of the landowner about the whole parcel. We also use the value willingness to sell to represent the intention of the landowner.

let WTS_{parcel} to be the willingness to sell value of a landscape parcel and WTS_{cell} to be the willingness to sell value of a landscape cell. We have

$$WTS_{parcel} = \frac{\sum_{cell \in parcel} WTS_{cell}}{\sum_{cell \in parcel} 1}$$

We use the average WTS_{cell} in a parcel to signify the willingness to sell a parcel.

Not only the internal factor influencing the decision of the landowners, but they could also be affected by their peers, such as their neighborhood, workmates or some financial agents. We use $peerinfluence$ to measure the outside influence on the landowners. In our study, we only consider the situation when the landowners are influenced by their neighborhood, specifically the other landowners who own the adjacent parcels.

Let P to be the peer influence value of one certain parcel, and we can calculate P with its adjacent parcels' WTS.

$$P = \frac{\sum_{parcel \in neighborhood} WTS_{parcel}}{\sum_{parcel \in neighborhood} 1}$$

The average WTS value of the neighborhood is regarded as the peer influence on a certain parcel, which can be interpreted as the more willingly the neighborhood to sell their land, the more likely a landowner would be to sell the land.

We have obtained the WTS of the landscape parcel and the peer influence to it, however, they have different dominants, we need to normalize the values before conducting more operations on them. The normalization method we use is as below.

Let $Normalized$ be the normalized variable, and x to be the original value.

$$Normalized = \frac{e^x}{1+e^x}$$

Hence, we have the normalized parcel WTS and peer influence; we plan to test how the urbanization differentials when given different proportions of parcel WTS and peer influence.

Let DV_{parcel} to be the determinative value for the landowner, WTS_{parcel} to be normalized value of the parcel's willingness to sell and P as the normalized value of the parcel's peer influence.

$$DV = \alpha * WTS_{parcel} + \beta * P$$

α and β are the coefficients we would give to the function when testing, indicating the proportion of the parcel's WTS and peer influence. For keeping the consistency of normalization, we simply should have the equation $\alpha + \beta = 1$.

With the normalized determinative value of the decision, we can set an arbitrary threshold for the landowners to make the ultimate decision.

Let $RAND$ be the random threshold we set for the landowner to make a decision, DV to be the determinative value of the decision and DTS to be the decision to sell.

$$DTS = 1, when DV \geq RAND$$

$$DTS = 0, otherwise$$

Human involvement simulating module passes the decision of the landowner to the urban growth model, assisting urban growth model to accomplish the urbanization simulation.

3.2.2 Agent-based Model

Agent-based modeling (ABM) is widely applied in the domain of dynamic simulation of the geographical system in the last decade. A definition is provided by Axtell: "An ABM consists of individual agents with states and rules of behavior. Running such a model is simply creating a population of such agents and letting agents interact, and monitoring what happens" [Axt00]. ABM is capable of decomposing a complex system into individual components with their attributes and behavior. The mechanism of ABM can be exploited through the simulation of human involvement during the urbanization.

Agent-based modeling is a technique for modeling dynamic systems from the bottom up. Individual elements of the system are represented computationally as agents. The system level behaviors emerge from the micro-level interactions of the agents. Therefore, it could depict the whole simulation procedure with the details of different levels, specifically reasoning how and when a decision is made by the landowners. Also, ABM aggregates all the information of its constituent elements and measure the outcomes in the system level over time.

There are several features of the agents in most of the agent-based models, which are briefly illustrated below[CH12]:

- Autonomy: all of the agents are autonomous entities, their functionalities are not affected by the system or other agents. The agents should have the ability to process the data internally and communicate with the environment and other agents through messages.

- Heterogeneity: the agents can be in different shapes of entities. Various agents maintain different sets of attributes and functions. Also, a group of the same kind agent is allowed as long as they have a similar set of attributes and functions.

- Active: all the agents contribute to the simulation outcomes.

All the agents together amalgamate the agent-based model, and different types of agents have different jobs in the simulation. The human involvement simulating module could be shaped into an agent-based model. And in this way, we can better interpret the relationships among different types of agents and observe the state change of agents.

three principal components to be documented about a model: Overview, Design concepts, and Details[Gri10].

3.3 Model Overview

3.3.1 Model Design Concepts

The key outcome of the model is the decision raster file, indicating if the landowner is willing to develop some certain landscape cell. The outcome synthesizes not only the landowners' condition but also their neighbors' impact on them. All of the landscape cells, parcel and landowner have adaptive behavior, as landscape updates development potential and landowner updates age after each time step, and landscape parcel updates its willingness to sell value once it senses the landscape cell holds a new willingness to sell value. And the landscape parcel is seeking the objective that the landowner is willing to sell this parcel. Stochasticity is used to determine the threshold to make the final decision. Also, a set of collective jobs is presented in the model, as the landscape cells within the same landscape parcel working as a group, their average value of willingness to sell is one of the landscape parcels, and they share the same DTS value; and the landscape parcel neighbors work in the shape of a group as well. Moreover, the agents interact with each other via messages.

3.3.2 Model Details

- Initialization

  Since all the raw landscape data is in the format of raster and the landowner profile is in the format of tables, we need to reassign all the data to corresponding agents.

- Input data

  Our human involvement simulating module is working with the urban growth simulation model, and the input data for each time step is passed in by urban growth simulating model as the updated development potential value of every landscape cell.

- Output data

  After each time step, the human involvement simulating module generates a raster of the landowner's decision on every landscape cell. The raster file indicates if the landscape cell in the corresponding position of the landscape earns the permission from its landowner to be traded for development into the urban area.

- Purpose

  The purpose of this model is to simulate the human involvement in urban growth procedure, especially the decision making of landowners in the land trade.

- Entities, state variables, and scales

  We briefly categorize the agents into three types: landowners, landscape cells and landscape parcels. And the state variables of each type of agents are listed below correspondingly:

  – Landowner

    Landowner ID: the unique identification of a landowner.

    Landowner Type: There are four types of landowners, we assume they would hold different attitudes when the land trade is happening. The main difference among the four types of landowner in computation is embodied in landscape cell's willingness to sell (WTS) value. The more conservative the landowner is, the larger the value is.

Landowner Age: the age of the landowner.

Landowner Income: the annual income of the landowner.

- Landscape Cell

  Landscape Cell ID: the unique identification of a landscape cell.

  Landscape Parcel ID: the id of the landscape parcel that this landscape cell locates in.

  Landowner ID: the id of the landowner who owns this landscape cell.

  Development Potential: the development potential value of this cell, this variable is estimated by urban growth model and passed in.

  WTS: value of willingness to sell by the landowner on this landscape cell.

  DTS: value of the decision to sell by the landowner on this landscape cell.

- Landscape Parcel

  Landscape Parcel ID: the unique identification of a landscape parcel.

  Landowner ID: the id of the landowner who owns this landscape parcel.

  Landscape Cell List: the ID list of the landscape cells which locates in this landscape parcel.

  Neighborhood Parcel List: the ID list of the landscape parcels which locates adjacent to this landscape parcel.

  Peer Influence: the value of the peer influence to this landscape parcel.

  WTS: value of willingness to sell by the landowner on this landscape parcel.

  DTS: value of decision to sell by the landowner on this landscape parcel.

• Process overview and scheduling

The communication among agents to agents or agents to the environment occurs via messages. Before introducing the functions of different agents, we define some messages in the communication.

- Message of Landowner Information

  sender: landowner

  receiver: landscape cells under the landowner's property

  content: the landowner's profile, including the landowner's ID, typology, age and income

- Message of Landscape Cell's WTS

  sender: landscape cell

  receiver: the landscape parcel which the sender in

  content: the landscape cell's ID and WTS

- Message of Landscape Parcel WTS

  sender: landscape parcel

  receiver: the landscape parcels which locates adjacent to the sender

  content: the landscape parcel ID and WTS

- Message of Landscape Parcel DTS

  sender: landscape parcel

  receiver: the landscape cells which locates into the sender

  content: the landscape parcel ID and DTS

3.3.3 FUTURES

We mentioned that the human involvement module is working with an urban growth prediction model, in our case, this urban growth prediction model is FUTURES (FUTure Urban-Regional Environment) [Mee13]. FUTURES is a multilevel modeling framework for simulating changing regional urban-rural divide based on observed land change patterns. It is a useful tool for researchers of urban systems to study the effect of changing landscape patterns with urbanization.

FUTURES considers a variety of environmental, infrastructural, socioeconomic and demographic factors as inputs to build a mathematical model representative of observed land growth patterns. These inputs are non-stationary drivers of land change influencing the spatial structure of new growth regions and known to vary with time. Also, FUTURES also considers simulated land conversion events in a feedback loop to predict new land change patterns. To incorporate the different drivers of land change, the spatial structure of land change events and their influence on further conversions, the FUTURES framework couples three interacting components: a DEMAND sub-model that projects the estimated demand for land in a region, a POTENTIAL sub-model that quantifies the site suitability of land in a region and, a patch growing algorithm (PGA) that produces projections of regional landscape patterns based on outputs from the DEMAND and POTENTIAL sub-models.

3.3.4 the FLAME

The FLAME(Flexible Large-scale Agent-based Modeling Environment) is an agent-based modeling framework, which aims to solve the problem caused by the complexities of implementing the simulation model to parallel computation architectures. This framework is incredibly robust and has been applied in a wide range of agent modeling domains that range from the biological science[Ric10] through to the modeling of the European economy [Dei08]. The agent-based model always faces the challenge of how to communicate among agents efficiently; the FLAME uses an MPI library as a message board library, thus making the framework is parallel and more flexible with message processing. The FLAME provides a bunch of functionalities to manipulate messages between agents, including the message filter and agent selector based on some certain conditions, which reduces the system working load and speeds up the whole simulation more.

With the structure of X-Machine of the agents, the FLAME can model the agents in a richer representation. The FLAME generates stimulating codes for the model using a template engine. All the agent definition should be formatted into a dialect of XML called XMML, and the template engine generates the simulation codes (Main.exe) along with the agent function implementation files. The initial agent configuration is also written in XML format and is passed to the program as a parameter to set up the initial agent states and population. The simulation code can also be executed on parallel computers using message passing interface (MPI) if configured so. Along with the initial configuration (0.xml), the simulation could generate the simulation results for each time step, which are all written in XML format. Figure 14 depicts the FLAME architecture.

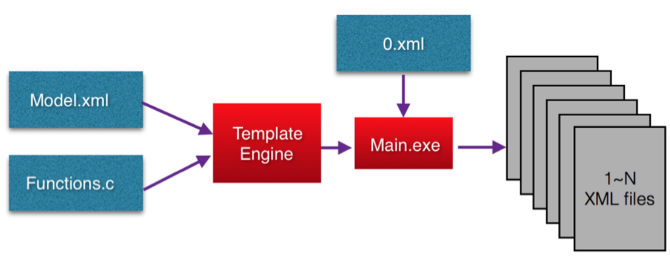

Figure 3.5 The logic flow of the FLAME

We have shaped our model into an agent-based model in the previous section; we make use of the MPI features by implementing the model with the FLAME framework.

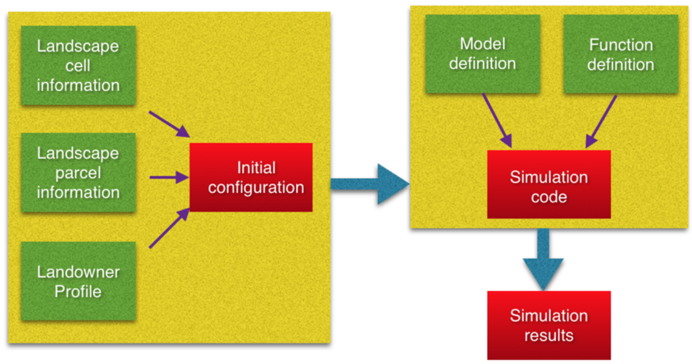

Figure 3.6 Flow illustration when implementing with the FLAME

First of all, we need to define the model into the specific XML format; we call this XML file âĂIJmodel definitionâĂİ. We depict the agent attributes, communication message details and the list of functions for landscape cells, landscape parcels, and landowners. Both model definition and the function implementations would be compiled to an executable file. In the meantime, we transfer the initial data from raw raster format to XML format. We pass the initial configuration file to the compiled executable simulation program, and the simulation result of each time step will be generated automatically.

3.3.5 Apache Spark

Apache Spark was started as a research project at UC Berkeley in the AMPLab, aiming to design a programming model which could accommodate more applications than MapReduce and maintain better fault tolerance. Apache Spark utilizes in-memory caching and optimized execution for fast performance, and it supports general batch processing, streaming analytics, machine learning, graph databases, and ad hoc queries. In short, Apache Spark is a cluster computing platform devised to be faster, more general and more robust.

The core concept in Spark is a resilient distributed database (RDD). An RDD is a distributed collection of data, with which the Spark workload can be directly expressed as creating a new RDD, transforming an existing RDD, or conducting operations on RDDs to generate a result. Spark distributed the data in the format of RDD across the cluster and parallelized the executions automatically.

The Spark is not designed for agent-based model specifically, however, it supports all the functions we need in an agent-based model. What we need to do is translate the functions in the model with Spark programming, basically creating RDD for each attribute of the agents and perform operations on them.

We assign an agent id to each agent, for instance, we number the landscape cell agents from the first to the last, as well as the landscape parcels and landowners, so we have #1 for landscape cell and 1 for landscape parcel. To identify the attribute of each agent, we define the RDDs in the tuple-like format as <Agent-ID,Attribute-Value>. For further development of the simulation module, we choose Python as the programming language for Spark since Python is integrated with a bunch of machine learning related library. We listed some agent-based model functions and involved Spark API.

- Read the initial data from raster

  we use parallelize() to read and distributed the data into RDD format, and sometimes combining with filter() and lambda() to sift the agent attribute value.

- Read the initial data from the CSV file

  we use textFile() to read a CSV file with RDD format, referred to RDD1, then parse all the attributes one after another by performing operations on RDD1 we use map() and lambda() to select the CVS columns which we are interested in.

- Pass the information from one agent to another

  In our model, there are several places where we need to pass one type agent information to another type of agent. For example, in the first steps of our simulation, the landowner should pass their profiles to the landscape cells which under his/her property, so we have <Cell-ID,Landowner-ID> indicating the ownership of the landscape cells and the landowner, and <Landowner-ID,Profile> as the landowner profile information. After the landowner passes the profile to the cells, we should obtain <Cell-ID,Profile>.

- Agent internal calculation

  we use mapValue() to transform state or calculate attribute value within an agent.

3.4 Experiment

We conduct the experiments on both the implementation with the FLAME and Apache Spark. With a set of a reliable data set, we test their performance differences on a computation cluster. And we also consider the situation when there is no cluster available for a researcher, so the experiments are conducted on a standalone machine as well. And in our experiment, human involvement module communicates with an urban growth model via system signal. Such loose coupling architecture enhances the flexibility of the system.

3.4.1 System Description

For our experiments, we have a cluster configured as 20 virtual machines with 2.4GHz dual-core, and the RAM size of each machine is 4GB. And we have two machines to test the standalone mode. One is a server-level Linux machine, it has 32 2.40GHz Intel(R) Xeon(R) CPU cores, with 128GB RAM installed. And the other is a personal use OS X machine, it has 2.6 GHz Intel Core i7, and 8 GB RAM installed.

3.4.2 Data Description

We use the data collected from Cabarrus County in North Carolina, and this county is very country and rural. It has many farm lands within the county and settled close to the broader economic city region. From the perspective of urbanization, the study of Cabarrus is significant.

The raw data we use is all in raster format, and the matrix is 1488*1025. We consider each element in the raster file as a landscape cell, also as an agent. Therefore, we have more than 1.5 million landscape cell agents. And we also consider landscape parcels as agents, and there are 78508 landscape parcels by statistical analysis. Also, there are 62830 landowners considered as agents. With all of these agents, we start the simulation experiments.

3.4.3 Experiment Results

All the experiments are done with both Spark and MPI environment. And since MPI performance is leveraged by the number of related CPU cores, in the experiment with the FLAME, we increase the number of involved cores from 1 to 20 to observe the performance difference caused by the core numbers.

With the server-level machine, we run the experiments 100 times for both MPI and Spark, and we calculate the average computation time of each scenario. The result of the FLAME environment is shown in Table 3.1.

We can see the trend of the computation time when involved computation cores in-

Table 3.1 Computation Time for Different Core Number with the FLAME Framework

Core Number	Time(s)	Core Number	Time(s)
1	216.33	11	266.38
2	220.22	12	268.79
3	227.13	13	272.38
4	233.04	14	276.00
5	241.00	15	283.27
6	245.09	16	292.63
7	250.28	17	336.06
8	253.99	18	348.90
9	258.04	19	364.58
10	262.14	20	375.80

creasing. More cores involve, more computation time spends, this is unexpected from the goal of MPI.

While the personal-use machine is not significant with CPU core numbers, so we only run the Spark environment for the experiment, the computation time of 100 runs is 15.45s. And figure 16 shows the landowner decision of the land trade in Cabarrus County.

The performance difference of the cluster is shown in Table 3.2.

3.5 Discussion

From the experiment result, we can see that the performance of the human involvement module implemented with Apache SPARK outperforms the one carried out with MPI. By analyzing the result and the framework, we have some understandings about the unexpected results.

Table 3.2 Comparison of Cluster performance

Node Number	MPI Times(s)	Spark Times(s)
1	418.87	21.72
2	223.85	36.32
3	593.50	38.87
4	540.87	34.48
5	537.02	36.94
6	563.83	36.96
7	552.63	36.89
8	568.85	36.00
9	561.94	35.32
10	553.92	34.69
11	558.03	38.29
12	586.14	38.47
13	590.23	38.11
14	584.65	37.87
15	585.10	37.43
16	620.20	38.54
17	575.33	39.21
18	610.92	37.38
19	622.98	37.17
20	643.20	37.09

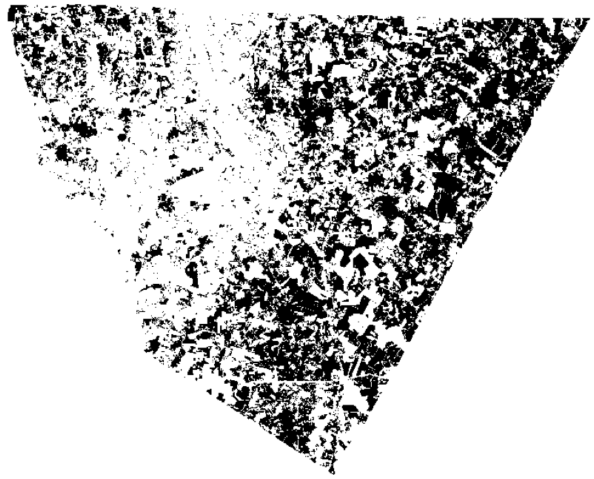

Figure 3.7 Landowners' Decision on land trade

3.5.1 Result Analysis

From the standalone and cluster results, the most outstanding finding is that we can see that the Apache Spark outperforms MPI in the experiments. And in the MPI standalone, we found that more threads were leading poorer performance. We dig into that problem and locate the reason for that is because the FLAME framework writes every agent instance to disk after each iteration, which requires lots of I/O operations, multiple threads extend the latency of the operation.

While in the cluster experiment, we also find some unexpected results, the computational time increases as more nodes added. We just divided the computational time into three parts when dealing with a cluster: data transfer time, actual computing time and synchronization time. With the node number increase, the time consumed by data transfer and synchronization take more parts of the total time. When the actual computation time decreases with more nodes, the other parts represent as dominant parts. Meanwhile, the time is affected by the data size. Obviously, we observed that the FLAME framework spends more time on data transfer than Apache Spark due to the significant size of its initialization file.

3.5.2 Limitations of the FLAME

THE FLAME Framework is designated to define an agent-based model, being capable of transiting the state of an agent from one to another. The transition functions have read and write access to all the agents and its attributes, as well as to the message board which is used in MPI features. This mechanism aims to keep the data consistency and better parallelize data when using MPI, however, in comparison with Apache Spark, we can view some limitation of the design with the benefit of hindsight.

Data Granularity: In the FLAME framework, all the operations performed on the agent is the unit of an agent, which means the smallest computation unit is the agent instance. In term of data partitioning for parallel computation, the FLAME treats an agent as the prime element. In the execution, we have to wait until the agent finishes all the read and write operation in one-time step, and then write the agent from memory to disk as a whole unit. This problem is apparently posed when dealing with the scalable size of data and complex agent dependency. In the meantime, the module implemented with Apache Spark crunches the agents into smaller pieces. The agent attribute is treated as the smallest data unit. The RDD in our design is the characteristics of the agent, and with the RDD transformation and operation, an agent could communicate with each other and change states internally or externally. Even with scalable size data, the data is distributed onto different nodes in the unit of agent attributes.

Message Board Usage: The message board usage is the core technique in the design of the FLAME framework. With the message board, the FLAME could leverage with the power of MPI to distribute the message between agents. However, the message board works in a role as a mediate layer, and all the agents send messages to the message board first, and all the agent read the messages from the synchronized message board. This operation involves both reads and writes access to the message board, considering a lot of agents, the design of message board brings heavy workload to the model. In our design of the module implemented with Apache Spark, we avoid the usage of the message board. The agent could communicate with each other via the agent attribute operations.

Disk vs. Memory: In the FLAME framework, the agent instance would be written to disk once it is finished the iteration. And the agents would be read into memory again when a new iteration starts. This causes the bottleneck of data access speed. While in Apache Spark, all the data is loaded into memory at the beginning of the execution, and only the wanted results are written to the disk. In our design of the module, we loaded all the land use information and landowner profile at first, and the data remains in memory during all the simulation process. The use of memory is the core technique of Apache Spark.

Other Implementation Concerns: In the implementation with both the FLAME and Apache Spark, the data pre-processing is also a key part of the simulation. The FLAME framework requires a very specific XML format of agent initial configuration file, and XML file operation is relatively complicated. For instance, in our experiment, it took 20 minutes to pre-process the initial configuration file for the FLAME framework with the data of Cabarrus County. And the time is sizable with the agent number, and the data pre-processing would be a greater concern when dealing with more agents.

In the Apache Spark implementation, our module simply read the raw raster files into memory and convert them into RDD at the meantime. In our experiment, the data loading took just three seconds.

Not only the data loading is time-consuming, but the outputting of the final result.

As mentioned above, XML file is complex to process with abundant tag information. We choose to generate separate raster files for different attributes of the agents. With the raster file, we can do further analysis with the location information provided by the raster matrix.

CHAPTER

4 CONCLUSION

Urban, rural and humans are tightly connected, human is the lever in the land use change. A better urban growth model could help planners to build a better habitat to live-in. In this thesis, we developed solutions to improve both the accuracy and speed of urban growth model.

In the first section (chapter 2), we explored the possibility of merging machine learning algorithms and agent-based model into the modeling of urban growth simulation. With the interaction between agent-based model and urban growth model, we simulated the decision-making process of landowners when dealing with the urban development. The multi-level decision approach outperformed the traditional binary decision approach.

Moreover, a reasonable bargain process between developer and landowner is shown to enhance the power of prediction.

In the second section, we implemented the ABM on apache spark platform to speed up the simulation of urban growth. We can see the agent-based model implemented with Apache Spark outperforms the one implemented with the FLAME(MPI). In our future work, we would like to further explore the following topics:

- Parcel Neighborhood Calculation

  The neighboring relationship in our study is based on landscape parcel, which is not a prime element in a location-based data format. We attempted to solve this in both the FLAME framework and Spark, but it is very time-consumptive since it involves Cartesian products. By reviewing the problem, we could use the data partition function in Spark to find the parcel neighborhood. Instead of processing all rows of the data, we can partition the data into different parts optimally with adjacent rows. Thus, the computational workload could be reduced exponentially.

- Visualization

  The agent-based model is always related to biology, economy, geology, etc., abundant choices of visualization could increase the readability of the results and a better understanding of the underlying details. We could like to improve the module with a built-in visualization module.

- Computational Steering

  Also, agent-based model could be used in a policy-making procedure with governments and different economical organizations, and it is likely that they want to interfere the simulating procedure with new policies, technically new state transition functions and new attributes values. Our implementation hasn't provided such functions yet, and with the memory use of Spark, it is possible to improve the module with computation steering functions.

- Generic Framework

  So far, we implemented the human involvement module in urbanization with Apache Spark, in comparison with the FLAME framework, we want to develop our module into a more generic usage framework. More scenarios and agent transitions should be taken consideration.

BIBLIOGRAPHY

[Ars13] Arsanjani, J. J. et al. "Integration of logistic regression, Markov chain and cellular automata models to simulate urban expansion". *International Journal of Applied Earth Observation and Geoinformation* **21** (2013), pp. 265–275.

[Axt00] Axtell, R. "Why agents?: on the varied motivations for agent computing in the social sciences" (2000).

[BH03] Barros, A. J. & Hirakata, V. N. "Alternatives for logistic regression in cross-sectional studies: an empirical comparison of models that directly estimate the prevalence ratio". *BMC medical research methodology* **3**.1 (2003), p. 21.

[Bat07] Batty, M. *Cities and complexity: understanding cities with cellular automata, agent-based models, and fractals*. The MIT press, 2007.

[Bre17] Breiman, L. *Classification and regression trees*. Routledge, 2017.

[CH89] Capozza, D. R. & Helsley, R. W. "The fundamentals of land prices and urban growth". *Journal of urban economics* **26**.3 (1989), pp. 295–306.

[CC10] Chen, B. & Cheng, H. H. "A review of the applications of agent technology in traffic and transportation systems". *IEEE Transactions on intelligent transportation systems* **11**.2 (2010), pp. 485–497.

[CZ14] Chen, C. P. & Zhang, C.-Y. "Data-intensive applications, challenges, techniques and technologies: A survey on Big Data". *Information sciences* **275** (2014), pp. 314–347.

[CG98] Clarke, K. C. & Gaydos, L. J. "Loose-coupling a cellular automaton model and GIS: long-term urban growth prediction for San Francisco and Washington/Baltimore". *International journal of geographical information science* **12**.7 (1998), pp. 699–714.

[Cla97] Clarke, K. C. et al. "A self-modifying cellular automaton model of historical urbanization in the San Francisco Bay area". *Environment and planning B: Planning and design* **24**.2 (1997), pp. 247–261.

[CV95] Cortes, C. & Vapnik, V. "Support-vector networks". *Machine learning* **20**.3 (1995), pp. 273–297.

[Cou97] Couclelis, H. "From cellular automata to urban models: new principles for model development and implementation". *Environment and planning B: Planning and design* **24**.2 (1997), pp. 165–174.

[CH12] Crooks, A. T. & Heppenstall, A. J. "Introduction to agent-based modelling". *Agent-based models of geographical systems*. Springer, 2012, pp. 85–105.

[DS14] Deep, S. & Saklani, A. "Urban sprawl modeling using cellular automata". *The Egyptian Journal of Remote Sensing and Space Science* **17**.2 (2014), pp. 179–187.

[Dei08] Deissenberg, C. et al. "EURACE: A massively parallel agent-based model of the European economy". *Applied Mathematics and Computation* **204**.2 (2008), pp. 541–552.

[Don08] Doniec, A. et al. "A behavioral multi-agent model for road traffic simulation". *Engineering Applications of Artificial Intelligence* **21**.8 (2008), pp. 1443–1454.

[dâĂ02] dâĂŹAquino, P. et al. "Agent-based models of land-use and land-cover change". *Proc. of an International Workshop*. 2002, pp. 4–7.

[Eas09] Eastman, J. R. "IDRISI Taiga guide to GIS and image processing". *Clark Labs Clark University, Worcester, MA* (2009).

[Fen16] Feng, Y. et al. "Simulation of dynamic urban growth with partial least squares regression-based cellular automata in a GIS environment". *ISPRS International Journal of Geo-Information* **5**.12 (2016), p. 243.

[Fil09] Filatova, T. et al. "Agent-based urban land markets: agent's pricing behavior, land prices and urban land use change". *Journal of Artificial Societies and Social Simulation* **12**.1 (2009), p. 3.

[FB97] Friedl, M. A. & Brodley, C. E. "Decision tree classification of land cover from remotely sensed data". *Remote sensing of environment* **61**.3 (1997), pp. 399–409.

[Gri10] Grimm, V. et al. "The ODD protocol: a review and first update". *Ecological modelling* **221**.23 (2010), pp. 2760–2768.

[Hal95] Hall, C. A. et al. "Modelling spatial and temporal patterns of tropical land use change". *Journal of Biogeography* (1995), pp. 753–757.

[Hua14] Huang, Q. et al. “A review of urban residential choice models using agent-based modeling”. *Environment and Planning B: Planning and Design* **41**.4 (2014), pp. 661–689.

[Jen96] Jensen, F. V. et al. *An introduction to Bayesian networks*. Vol. 210. UCL press London, 1996.

[KC03] Kalnay, E. & Cai, M. “Impact of urbanization and land-use change on climate”. *Nature* **423**.6939 (2003), p. 528.

[KZ01] King, G. & Zeng, L. “Logistic regression in rare events data”. *Political analysis* **9**.2 (2001), pp. 137–163.

[LL07] Li, X. & Liu, X. “Defining agentsâĂŹ behaviors to simulate complex residential development using multicriteria evaluation”. *Journal of environmental management* **85**.4 (2007), pp. 1063–1075.

[LY00] Li, X. & Yeh, A. G.-O. “Modelling sustainable urban development by the integration of constrained cellular automata and GIS”. *International Journal of Geographical Information Science* **14**.2 (2000), pp. 131–152.

[LW02] Liaw, A., Wiener, M., et al. “Classification and regression by randomForest”. *R news* **2**.3 (2002), pp. 18–22.

[Mag11] Magliocca, N. et al. “An economic agent-based model of coupled housing and land markets (CHALMS)”. *Computers, Environment and Urban Systems* **35**.3 (2011), pp. 183–191.

[Mat07] Matthews, R. B. et al. “Agent-based land-use models: a review of applications”. *Landscape Ecology* **22**.10 (2007), pp. 1447–1459.

[MS00] Mayer, C. J. & Somerville, C. T. “Residential construction: Using the urban growth model to estimate housing supply”. *Journal of urban economics* **48**.1 (2000), pp. 85–109.

[McM96] McMillen, D. P. “One hundred fifty years of land values in Chicago: A nonparametric approach”. *Journal of Urban Economics* **40**.1 (1996), pp. 100–124.

[Mee13] Meentemeyer, R. K. et al. “FUTURES: multilevel simulations of emerging urban–rural landscape structure using a stochastic patch-growing algorithm”. *Annals of the Association of American Geographers* **103**.4 (2013), pp. 785–807.

[PF08] Parker, D. C. & Filatova, T. “A conceptual design for a bilateral agent-based land market with heterogeneous economic agents”. *Computers, Environment and Urban Systems* **32**.6 (2008), pp. 454–463.

[Par02] Paruchuri, P. et al. “Multi agent simulation of unorganized traffic”. *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*. ACM. 2002, pp. 176–183.

[PJ01] Pontius Jr, R. G. et al. “Modeling the spatial pattern of land-use change with GEOMOD2: application and validation for Costa Rica”. *Agriculture, Ecosystems & Environment* **85**.1-3 (2001), pp. 191–203.

[Ric10] Richmond, P. et al. “High performance cellular level agent-based simulation with FLAME for the GPU”. *Briefings in bioinformatics* **11**.3 (2010), pp. 334–347.

[Rom99] Romero, H. et al. “Rapid urban growth, land-use changes and air pollution in Santiago, Chile”. *Atmospheric Environment* **33**.24-25 (1999), pp. 4039–4047.

[Sha16] Shashidharan, A. et al. “pFUTURES: A Parallel Framework for Cellular Automaton Based Urban Growth Models”. *The Annual International Conference on Geographic Information Science*. Springer. 2016, pp. 163–177.

[SC02] Silva, E. A. & Clarke, K. C. “Calibration of the SLEUTH urban growth model for Lisbon and Porto, Portugal”. *Computers, environment and urban systems* **26**.6 (2002), pp. 525–552.

[Sud04] Sudhira, H. et al. “Urban sprawl: metrics, dynamics and modelling using GIS”. *International Journal of Applied Earth Observation and Geoinformation* **5**.1 (2004), pp. 29–39.

[War00] Ward, D. P. et al. “A stochastically constrained cellular model of urban growth”. *Computers, Environment and Urban Systems* **24**.6 (2000), pp. 539–558.

[Wat08] Watkiss, B. M. “The SLEUTH urban growth model as forecasting and decision-making tool”. PhD thesis. Stellenbosch: Stellenbosch University, 2008.

[Yan08] Yang, Q. et al. “Cellular automata for simulating land use changes based on support vector machines”. *Computers & geosciences* **34**.6 (2008), pp. 592–602.

[Yu09] Yu, L. et al. “An intelligent-agent-based fuzzy group decision making model for financial multicriteria decision support: The case of credit scoring”. *European journal of operational research* **195**.3 (2009), pp. 942–959.

[Zha16] Zhang, Q. et al. “Agent based urban growth modeling framework on Apache Spark”. *Proceedings of the 5th ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data*. ACM. 2016, pp. 50–59.

[Zhu08] Zhu, S. et al. *Agent-based route choice with learning and exchange of information*. Tech. rep. 2008.